OpenPose based Gait Recognition using Triplet Loss Architecture

MSc Research Project
Data Analytics

Shahil Shaik
Student ID: x18131743

School of Computing
National College of Ireland

Supervisor: Dr. Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shahil Shaik
Student ID:	x18131743
Programme:	Data Analytics
Year:	2019
Module:	MSc Research Project
Supervisor:	Dr. Vladimir Milosavljevic
Submission Due Date:	12/12/2019
Project Title:	OpenPose based Gait Recognition using Triplet Loss Architecture
Word Count:	6241
Page Count:	18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	30th January 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

OpenPose based Gait Recognition using Triplet Loss Architecture

Shahil Shaik
x18131743

Abstract

Behavioural biometrics have certain advantages over physiological biometrics. These biometrics does not require cooperative subject or proximity of individual. Gait recognition is form of behavioural biometrics where individual is identified based on their walking pattern. Most of the gait recognition models are based on temporal templates of human silhouettes such as gait energy image. In this project gait recognition using the OpenPose key point coordinates is proposed. OpenPose algorithm estimates the coordinates of various key points of individual in an image or video. Two approaches for feature extraction is applied on CASIA-B dataset. Manual features such as length of limbs and angle between limbs of individual are extracted using OpenPose key points coordinates. A 64-dimensional embedding vector is created for every video using deep learning triplet loss algorithm. Knn algorithm is trained on manual features and triplet loss features separately. An accuracy of 65 percent is achieved using the manual feature method and 71 percent using the triplet loss architecture.

1 Introduction

Biometrics are used for identification of an individual in various sectors. Biometrics such as facial recognition, fingerprint and Iris are used as identification metrics to enhance security. These are even used to enhance security for personal devices like smartphones or laptops by preventing access to unidentified individual. Biometrics are majorly classified as physiological and behavioural biometrics. Physiological are based on physical features of the individual such as fingerprints, facial recognition and iris, whereas behavioural biometrics are based on individual behaviour patterns such as Gait and voice. Though the accuracy of identifying individuals using physiological biometrics is high, there are certain limitations to it. These methods require cooperative subject and a proximity for identifying, whereas behavioural biometrics do not need a cooperative subject for identification. Gait is a type of behavioural biometric where a persons manner of walking is used for identification. Such biometrics can be helpful for a forensic team in identifying the convict from the suspects. For example, a video of the robbery can be used to identify the robbers using their gait captured in the video. Physiological biometrics fail in scenarios where robbers utilise gloves and face mask. Most of the gait recognition algorithms currently available are based on Gait energy image. It is a collection of silhouettes of walking cycle and aligning all frames into single image.

OpenPose algorithm is used for estimating the pose of the persons from an image or video. Coco had organised a key point detection challenge where the goal is to create an algorithm to identify the key points of the human pose using image or video. Coordinates of key points such as feet, knees, elbow, shoulders, hips and so on should be obtained using the image. A new algorithm is introduced by CMU called OpenPose won the challenge. This algorithm has recently been incorporated in the OpenCV library. OpenCV is an open source library of Python which has various computer vision related algorithms in it. In this paper, coordinates of the key points obtained using OpenCV library are used in place of gait energy image for gait recognition. The key points of all the frames in walking cycle are considered for creating a gait recognition algorithm. Two different approaches are used for creating the algorithm, namely, manual feature creation and dynamic feature selection using triplet loss architecture. In manual feature selection different features such as length of arms, legs, max angle between legs and length of arm stroke are calculated using the key point coordinates. These features are used to create nearest neighbour and similarity check algorithms like Euclidian distance to identify individuals based on features. In dynamic feature selection, a neural network algorithm is used with triplet loss function. This algorithm is trained to create dynamic 64-dimension vector that are clustered based on class. A knn algorithm is trained on these data for identification of individuals.

2 Related Work

2.1 Gait Recognition

Gait recognition is achieved by using various types of data such as lidar, multicamera human model, footstep pressure and gyroscope. These methods can be grouped into two types, namely, model-based (Guoying Zhao et al.; 2006) and appearance based (Goffredo et al.; 2010). The model based method uses a 3d model of individual that is created using various methods (Ariyanto and Nixon; 2011). In appearance-based algorithms human silhouettes of the gait cycle are extracted from the video and a single superimposed image of these silhouettes is created using various methods. The most popular form of such image is called Gait Energy Image (GEI) which is discussed in the research (Ju Han and Bir Bhanu; 2006). All the silhouettes of a walking cycle are normalized to the same size and these normalized silhouettes are aligned horizontally by centring the upper half of silhouettes based on its horizontal centroid. These aligned silhouettes are time normalized to form a Gait energy image. GEI of different individuals are used as an input to various similarity check algorithms such as Euclidian distance, to create the gait recognition algorithm. Different variations of these gait energy images are available such as Motion Energy Image (MEI) and Motion History Image (MHI). Motion Energy Image is a cumulative of binary motion images computed from the start frame to the last frame. The shape of the region obtained using this process can be used to suggest the viewing condition and motion of objects. Motion History image is used to represent the motion of the object in the frame. Pixel intensity is defined as a function of temporal history of motion at a given point. This gives us image in which pixels of most recent movement are brighter. Combination of MEI and MHI are used in research (Bobick and Davis; 2001). Some of the limitations of these methods is the presence of a moving object other than the human in the frame.

Extraction of features from the silhouettes and GEI have been brought to focus in the study conducted by (Yaacob and Tahir; 2012) and team. Discrete Cosine Transform (DCT) is used for feature extraction and Principle Component Analysis (PCA) for feature selection. DCT is used for feature extraction from average silhouette and max width stride. PCA is used on these features for dimensionality reduction and to select the components with most variance. These selected features are trained on Artificial Neural Network (ANN) classifier to identify the individuals.

Supervised and unsupervised feature selection from GEI is discussed in the paper (Khalid Bashir et al.; 2008). A cross validation wrapper algorithm is adopted for supervised learning where data is divided into validation and training set. Subset of features are selected by this algorithm to obtain the optimum classification rate on the validation set. In unsupervised feature selection, the standard deviation of the GEI intensity across all GEI templates for all pixel locations are calculated. These scores are used as features. A threshold is defined to classify the individuals based on the extracted features.

Difference in view affects the accuracy of feature extraction of gait. A method to solve this problem is focused in (Li et al.; 2017). Low rank optimization is used for view normalization of the gait. By using background difference method, width of the gait is detected based on the change in silhouettes. Gait frame difference entropy image is extracted using these widths. Augmented lag-range multiplier method is used on these images for low rank optimization, wherein, the nearest neighbor classifier finally classifies the optimized image.

In order to solve the difficulties faced by change of view, a new view-invariant feature for cross-view gait recognition is discussed in the paper(Kusakunniran et al.; 2013).Unlike previous paper view normalization is performed at input layer. All the silhouettes from different view are transformed to a common canonical view with the help of low rank textures. A gait description based on Procrustes Shape Analysis (PSA) is applied on these canonical views to recognize gait. A new invariant feature of gait is extracted using Procrustes mean shape. Procrustes distance is used for measuring gait similarities. Though the process showed desired result, it only has a view transformation range between 54 degrees to 126 degrees. A complete Multiview gait recognition is obtained in the paper (Kusakunniran et al.; 2009). A view transformation model is created by using Singular Value Decomposition (SVD) on GEI. Linear Discriminant Analysis (LDA) is used to further improve the performance of VTM and optimize GEI feature vector. Euclidean distance is adopted for the similarity measurements of gait. Gait signature obtained after optimized VTM model is checked with this similarity model for gait recognition. Though this process addresses all transformation angles, a higher accuracy of 90 percent is obtained in only transformation up to 18 degrees difference.

A discriminant approach for cross view gait recognition is proposed in the analysis performed by (Mansur et al.; 2014). Multiview discriminant analysis (MvDA)is used for achieving this. MvDA works by creating common discriminative subspace where variation between classes is maximized and within class is minimized. The output from the MvDA for different classes can be linearly separable. Eigen value distance is used for recognition of gait using MvDA output. Limitation of this method is bound with limitations of the MvDA process. MvDA process is sensitive to the dimension of data. Multiple sensitivity analysis revealed that errors increases as the number of training subjects decrease.

A more advanced technique called generative adversarial network is used to extract invariant gait features in paper (Yu et al.; 2017). These features obtained can handle different variation in gait data such as view angle, clothing and posture. A GAN model is used to generate normal clothing side view angle and without bag image. One of the advantages of this model is that view angle and other variation information is not required for generation of the normal gait. Two discriminators are used in this architecture unlike regular GAN. One of them is for fake/real and another for human identification information. The architecture of GaitGAN has an encoder-decoder generator and two discriminators. Encoder decoder are trained with help of two discriminator such that output of the decoder is a normal right-angle gait for any variant of gait. The result of this method is similar or in few cases less than other state-of-the-art algorithm. Deep CNNs are first used for gait recognition in paper (Wu et al.; 2017). Unlike the regular similarity check algorithms such as Euclidian distance and cosine similarity,

similarity learning by deep CNNs is used for identification of human beings based on their gait. Empirical evaluation of different scenarios such as cross walking conditions and cross view is provided with help of different network architectures and preprocessing approaches. Three different network architectures are discussed in this process. Two Gaits one of probe and another from gallery are given as input to these networks after few convolution, spatial max pooling and normalization layer. A final softmax layer predicts if both the probe and gallery gait are similar. This method outperforms other state-of-the-art models by a significant margin.

Gait of a same person may vary in some scenarios such as carrying a heavy object or looking down at a phone. A deformable registration model is introduced in paper (Makihara et al.; 2018) to tackle intra subject posture changes. A deformation field is computed to minimize difference between a gallery and the probe morphed deformation field. Eigen deformation modes of intra subject are learned which are different from inter subject deformation modes. This output can be combined with various types of gait recognition algorithms to increase the gait recognition accuracy for posture changes. This free from deformation model combined with a recent deep learning framework has improved the discrimination capabilities. Testing on 1334 subjects showed an effective improvement compared to direct matching of gaits.

In paper (Cheema et al.; 2012) instead of using sequence of temporal templates to model gait pattern, contour distance feature and key pose learning approach is used for gait recognition. A non-temporal collection of key poses are used for modelled as gait patterns. This paper shows effectiveness of using methods other than temporal templates for gait recognition. A 2D stick figure with key points extracted from silhouette guided by anatomical knowledge is used for creating a gait recognition algorithm in paper (Yoo et al.; 2008). 2D figures are created on the temporal sequence of silhouette to represent a gait signature and are used as feature for gait recognition. These features are trained over a back propagation neural network to recognize individuals. Out of 27 parameters of gait features, only 10 are selected based on their classification importance. A recognition rate of 90 percent is obtained for the test over 30 subjects. The First step in achieving pose-based gait recognition is detection of gait cycle using pose. This is achieved in the paper (Shen et al.; 2019). Latest pose estimation algorithm is used over a video sequence of person walking to get the human skeleton and coordinates of key points. Distance between the feet is used to create a gait cycle. A few outliers from the sequence due to false estimation of few key points are handled by defining a metrics called Ratio of Max to second (ROMS) which is the ratio of max amplitude to the second. After handling outliers, they calculated gait cycle by means of Fourier transformation.

Gait recognition with the help of pose estimation algorithm is discussed in the paper (Sokolova and Konushin; 2019). Motion of points in the area of human joints is used as a feature. Optical flow between frames is used to extract motion information. Various network architecture, body parts and aggregation methods are analyzed to get the optimum working combination. A CNN based network is used followed with PCA and average pooling for creating a gait recognition algorithm.

2.2 Pose Estimation

Combination of local observations on body parts and their spatial dependencies are used to infer a human pose estimation in previous years. Models created using these spatial dependencies can be classified as tree base models (Andriluka et al.; 2010) and non-tree base models. One of the tree-based models is discussed in paper (Ramanan et al.; 2005). Their algorithm assumes that certain canonical poses are taken by people even while performing unusual tasks such as skating and throwing a ball. A discriminative appearance model is build using limbs estimated from the detection. As per the paper features that discriminate a human figure in a frame will discriminate in the other frames as well. This model is used to detect the limbs from a figure in unrestricted poses. They can track multiple people in a video successfully. Multi view body pose estimation algorithm to operate in uncontrolled environments with low resolution is created by (Germann et al.; 2011). It is achieved by following two steps. Extracting the body pose using the spatial temporal silhouette matching for each camera and guessing the triangulated 3D pose. This estimated pose may have some ambiguities flip of symmetrical parts. To handle this an optical flow-based technique is used to detect a consistent sequence. The resulting 3D skeleton matches silhouettes from all views.

Convolution neural networks is used for creating a pose machine framework to learn various image dependent spatial models and image features in paper (Wei et al.; 2016). Dependencies between the variables for articulated pose estimation is modeled in this paper. Belief maps from previous frames are fed to a sequential convolution neural network to produce a refined estimate of part location. They also addressed the difficulty of vanishing gradient with the help of natural learning objective function. Heatmap of the image is used as an input for CNN in paper (Bulat and Tzimiropoulos; 2016). It has a two-part architecture where a set of N part heatmaps is obtained from the first subnetwork in which individual body parts are detected using per pixel sigmoid loss. Heatmaps obtained are sent to regressor subnetwork where heatmaps are staked along with image to confidence maps body part representation. Cascade proposed in this paper is flexible enough to integrate with other CNN architectures. An adversarial network with two discriminator and a multipose generator is used for pose estimation in paper (Chen et al.; 2017). Reasonable poses are distinguished form unreasonable poses by the two discriminators. Multitask pose generator uses this discriminator as an expert that distinguishes real and fake pose and trains to create a pose that deceives the expert as real. This process creates more effective pose estimation that can handle overlapping, occlusions and twisting of human bodies. This method can also be applied to other shape estimation problems such as detection of face landmark using DCNNs. A different approach to pose estimation by using compositional model is discussed in paper (Tang et al.; 2018). Hierarchies of meaningful parts and subparts are represented by compositional models. They also provide high order relationships among body parts which helps to resolve low level ambiguities. Introducing deeply learned compositional model solves the problem with prior models in handling complex cases.

3 Methodology

The objective of this project is recognition of gait patterns of an individual and to create a knowledge base of gait features. To obtain the knowledge of gait recognition several data mining steps such as data cleaning, transformation and data modelling are performed. Based on the above-mentioned requirements, a well-known data mining methodology for pattern recognition called Knowledge Discovery in Databases KDD is selected as research methodology. As shown in Figure 1 KDD has five main steps namely data selection, data pre-processing, data transformation, data modelling and evaluation. The pre-requirements of the transformation and input for data model are uncertain in this research. Data pre-processing and transformation may change based on the results obtained at the evaluation part. In this methodology changes can be made to pre-processing and transformation of data at any given step in order to achieve better performance in evaluation.

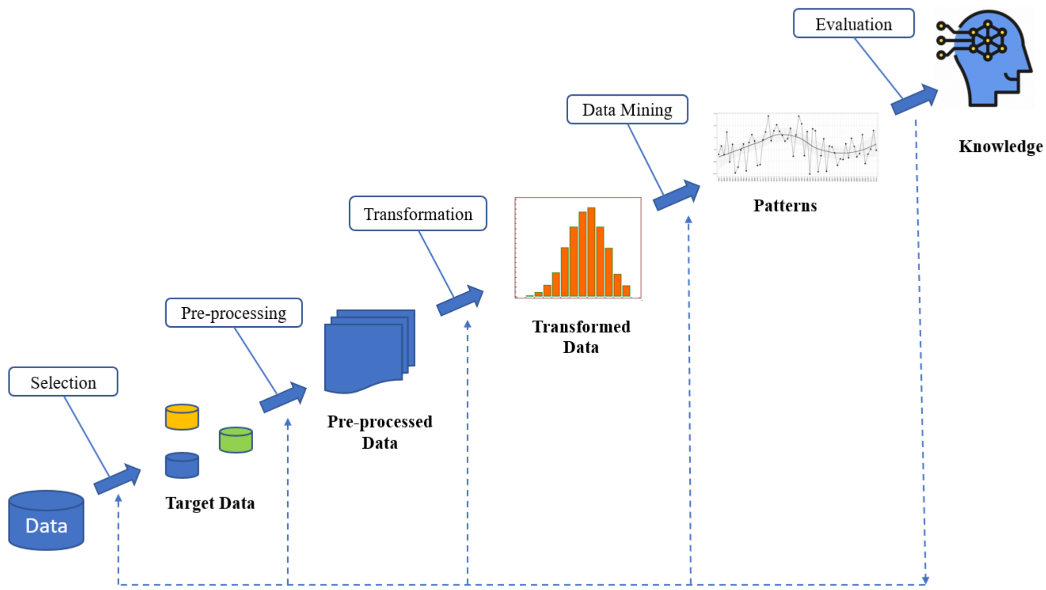

Figure 1: KDD Methodology

• Data Selection

In order to achieve an optimum and reliable gait recognition, we need a well-known benchmark gait database. CASIA-B provided by Chinese centre for biometric research is used as a benchmark data in most of the gait related research papers. It has walking videos and silhouettes of 124 subjects in various conditions.

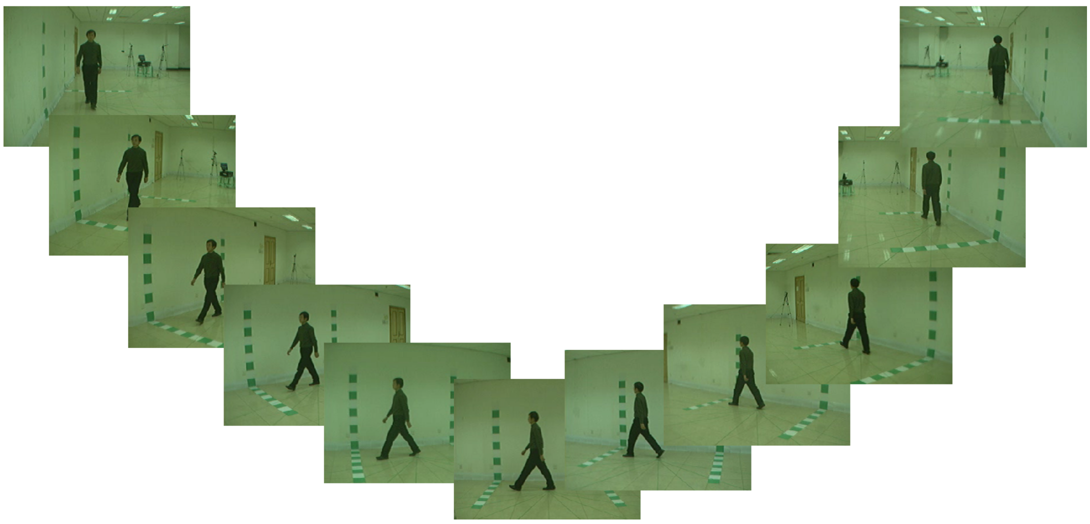

Figure 2: CASIA-B video data in 11 different views

Video data is considered for this research. It has videos in .avi format with total combined size of 10.8GB. walking of all 124 subjects is collected in 11 different angels. Videos of three walking conditions of each subject namely normal, bag and coat are present in data. Each video follows the naming format of xxx-mm-nn-ttt.avi in which ttt is camera view angle, nn is a sequence number every variation has six videos. mm represent walking condition nm for normal, bg for bag and cl for coat. For this research only the normal videos in 90-degree angle are considered. Figure 2 shows various frames in a video in 11 different angles

- **Data Pre-processing**

Key points from the walking subject for every frame in the video are obtained using OpenPose algorithm. OpenPose is recently incorporated in OpenCV library of python. With the help of this library and using COCO Caffe model weights, key point coordinated are estimated. A 256*256 grid is used for estimation of coordinates. Frames without the subject or incomplete subject are removed. All the missing values are filled with the previous values. Outliers obtained in the key point due to ambiguity and low resolution are removed.

- **Data Transformation**

An array of key points is obtained after the pre-processing. These key points are used to create new features such as length of the limbs, distance of stride, distance between feet, max angle between legs. Array of key points is reshaped according to the requirement of the data models.

- **Data modelling**

A simple nearest neighbour algorithm is trained on the manual features obtained for each individual video in data transformation phase. An algorithm based on neural networks, the triplet loss algorithm is trained in order to create dynamic embedding of key point such that the embedding of different individual is linearly separable. The nearest neighbour and similarity check algorithms are trained on this embedding output obtained from deep neural network.

- **Evaluation**

Metrics like f1-score, accuracy and precision of the nearest neighbour algorithm for both the models are evaluated to check their performance. In order to check their performance for similarity measurement Rank 1 and Rank 5 accuracy is measured. The deep learning triplet loss algorithm is evaluated based on the validation loss for each epoch. A comparison of these evaluation metrics for all variations of model is performed to obtain the effective algorithm.

4 Design Specification

In order to achieve gait recognition using OpenPose two types of models are proposed in this paper. Extraction of features such as length of leg, stride and angle between legs to create a nearest neighbour algorithm to predict the individual. Training the sequence of key points in a gait cycle using a triplet loss deep learning algorithm to obtain an embedding that is linearly separable between different individuals. OpenPose algorithm is used initially to extract the 14 key points from every frame in all video. These key points are processed and reshaped as per the requirements of the predictive models. Finally, performance of models is evaluated

Architecture of the proposed implementation is shown in Figure3 It has various stages such as data filtering, keypoints extraction from OpenPose, Manual feature extraction, Dynamic feature extraction using triplet loss architecture, Knn for class prediction and Evaluation. Each stage is explained in detail in below sections.

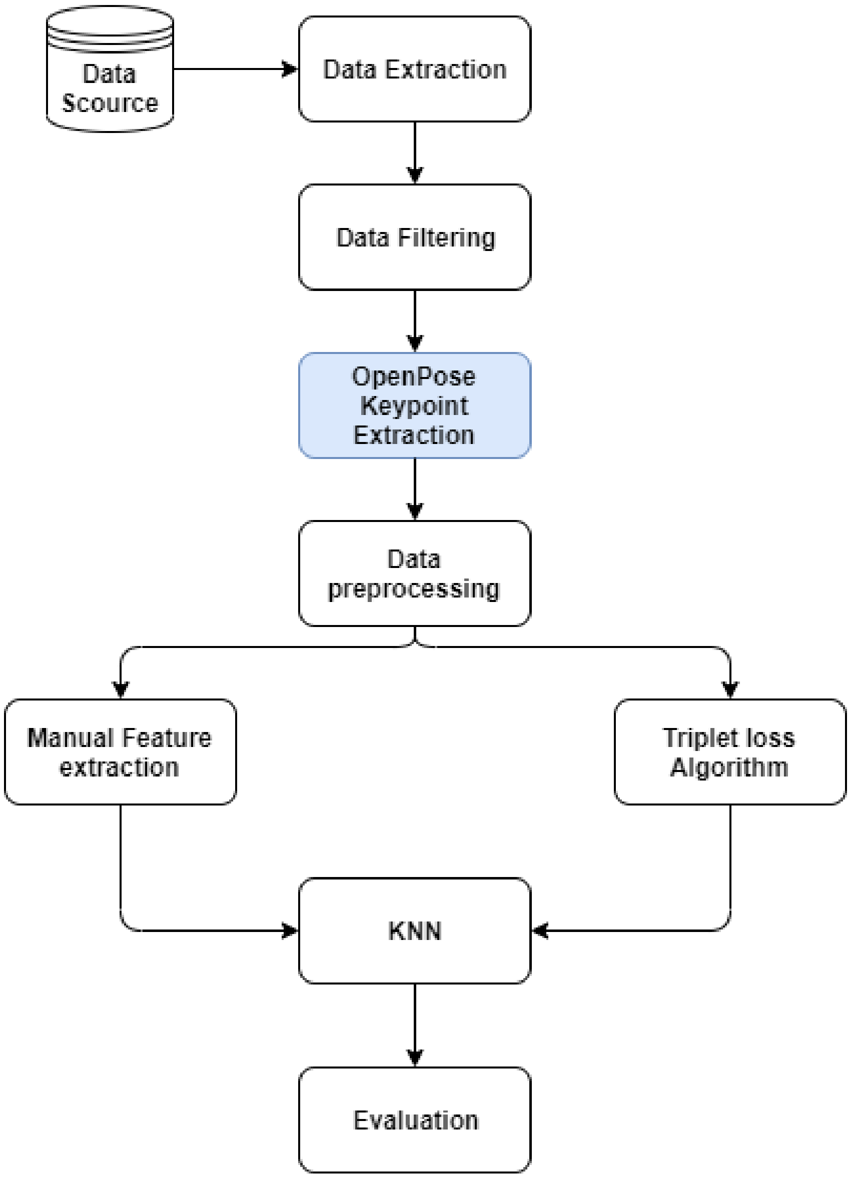

Figure 3: Architecture of project implementation.Coloured part indicate novelty of project

- **Data filtering**

Data set used for this project has videos of total size 10.8GB. It has data of 124 subjects in 11 views and 3 modes. Based on the time line of the project, only the videos of 90 degree view and normal mode are utilized for model training. Initially all the videos of normal gait in 90-degree angle is filtered with the help of glob package in python and stored in a separate folder for processing.

- **OpenPose**

OpenPose is a state of art key point estimation algorithm. Using OpenPose is a novelty of this project. Unlike other gait recognition algorithms based on silhouette template, OpenPose key points are used in this project for gait recognition. OpenPose gives the coordinates of 18 key points such as feet, knee, elbow shoulder and so on. This approach is the winner of coco key point detection challenge. It processes every frame of the video. It checks for probability of a key point at every point of a point grid on the image, In our case 256*256 grid. Then the points with probability more than required threshold are filtered and joined to form a 2d skeleton of the individual. Coordinates of the key points is saved as vector.

- **Manual Feature Extraction**

Once all the key points for every video is extracted. A list of features that can be unique to every individual gait are selected based on the previous research. Features of the individual such as length of leg, hands, angle between limbs and length of stride are manually calculated with the help of the coordinates acquired from openPose algorithm. Math function in library are used for calculation distance between two coordinates and angle between three coordinates. A vector of these features are created for every video and saved in a csv file.

4.1 Triplet Loss

In a normal deep learning classification problem, number of classes are predefined and are finite, whereas for a gait recognition problem number of classes is based on the number of subjects. Creating a normal softmax classification will be limited to the identification of only trained individual. Training the network for every new subject is not a reliable solution. In such scenario triplet loss is used to create an intermediate embedding representation such that contextually similar data points are projected in a near by region and dissimilar points are projected in far away from each other in a high dimensional vector space.

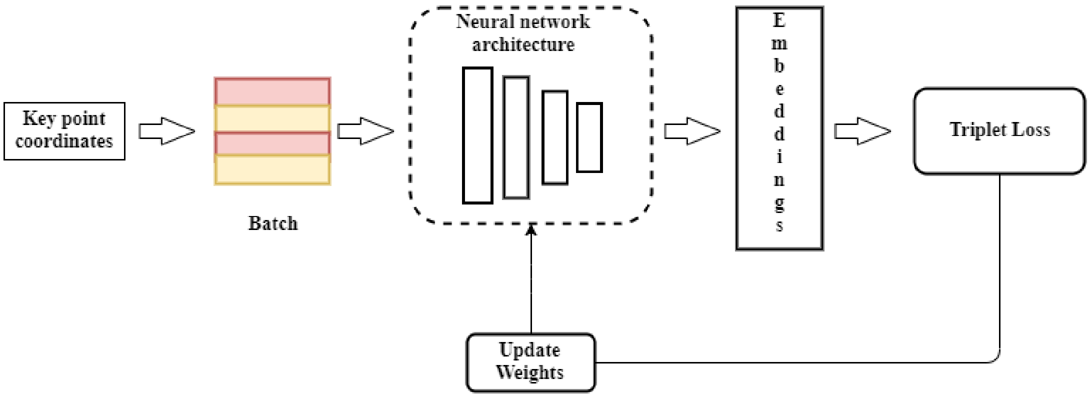

Figure 4: Triplet loss architecture

A triplet loss architecture has three identical networks namely anchor, positive and negative having similar neural net architecture and shared weights. A N-dimensional vector is created at the last layer of deep network. Three data points namely anchor, positive and negative are passed to network for updating the weights. Model learns to cluster the data points of same and also similar gaits in neighbouring region. Cost function used for triplet loss is shown in equation 1. d represent the distance between the points. Aim of this function is to reduce the distance between anchor and positive and increase the distance between anchor and negative.

$$\mathcal{L} = \max(d(a,p) - d(a,n) + \text{ margin }, 0) \quad (1)$$

Architecture of the deep learning network is shown in Figure4. Key point coordinates from OpenPose are reshaped as required. Batch of these coordinated are fed to neural network for training. These neural network outputs embedding vector for every video in the batch. Triplet loss is calculated for these embedding using the formula in equation 1. Weights of the neural network are updated using back propagation to reduce the loss value.

• KNN Algorithm

Nearest neighbour algorithm is used for the final identification of individual in both the approaches. K nearest neighbour classifies the given data point based on its k nearest points in the dimensional space. Manual features extracted from the key points are trained on nearest algorithm. Embedding vector obtained from the triplet loss network is also trained on nearest learning algorithm for identifying the individual.

5 Implementation

5.1 OpenPose key point extraction

Extraction of key points using OpenPose can be achieved with help of programming language such as JavaScript, C++ and python. Since python is chosen as the programming language for implementation of the model, Python library for OpenPose is used for key point extraction as well. OpenPose is recently incorporate in famous open source computer vision library OpenCV. OpenCV library is installed using pip-install. Pretrained weights and model for COCO data is downloaded and loaded using the library. This model will give 14 key points for a frame as shown in the Figure5. All the videos used for this process is filtered and stored in a folder. Using OpenCV Video capture and frame read. Every frame of each video is processed, and key points are estimated. A list of key points for every frame in a video is stored in a variable. This list is stored as a pickle file for further use.

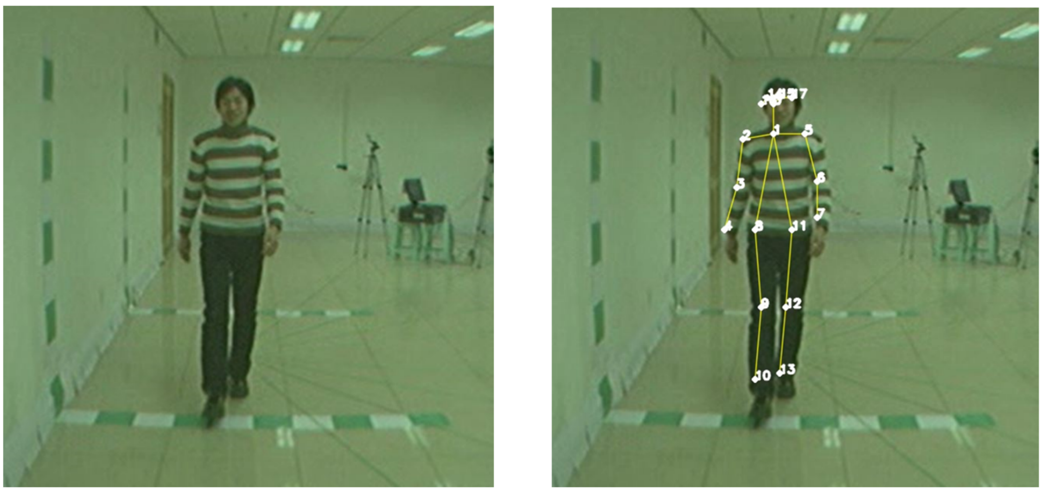

Figure 5: Key points detected using OpenPose

5.2 Feature Extraction

In this section features are extracted manually for creating a gait identification model. Along with the 14 key points extracted from OpenPose, few other metrics are calculated using key point coordinates. A function to calculate the angle between three points is created. Using this function angle between the legs, the angle between torso and hands is calculated. Length of upper body, leg, and hand are measured in every frame. All the 14 points and 6 additional features created are analysed for a gait cycle of the individual. Using these 24 features for every frame in a video, few aggregated features for an individual in the video are created for gait identification. Features such as maximum angle between legs during a gait cycle, amplitude of neck, hip in a gait cycle, length of swing of legs and hands are calculated. For a given video, features are stored in a vector and the list of these vectors for all videos are saved as a pickle file. A sample gait cycle for a key point is shown in the Figure6. Due to the low resolution of video a step wave is created in place of a proper wave. Few outliers are created in the cycle due to ambiguity created by low resolution during key point estimation by OpenPose. These outliers are removed, and features are calculated on the clean wave. Features such as wave length and amplitude are calculated for various key points.

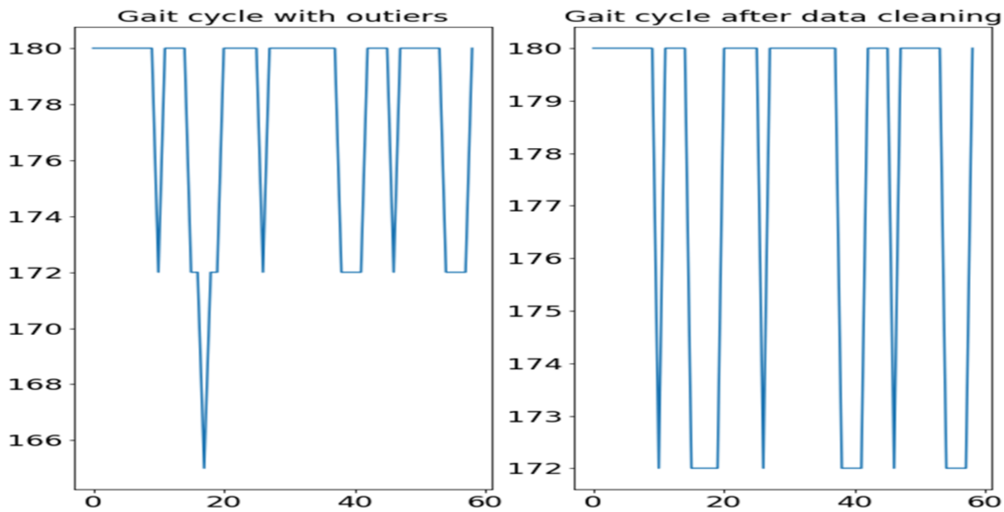

Figure 6: Gait cycle of a key point before and after outlier removal

5.3 Triplet Loss architecture

Keras with tensorflow as backend is used to create a deep neural network architecture. Initially, a base network is created with sequence of Dense and dropout layer starting with flatten. As shown in Figure7 bases network has flatten layer which flatten the input data of size 30*28, a dense layer of 128 neurons with activation as relu, a dropout layer with 10 percent dropout followed by layer of dense with embedding size of 64. Based on the base architecture, a triplet loss network is formed with model input as the vector of input data, input label and output as input labels, embeddings. The adam optimizer with learning rate of 0.0001 is used. A triplet loss function adopted from TensorFlow triplet loss function. Model is trained for 10000 epochs with batch size of 128. The final weights are loaded into the model and embedding vector of length 64 is created for every video.

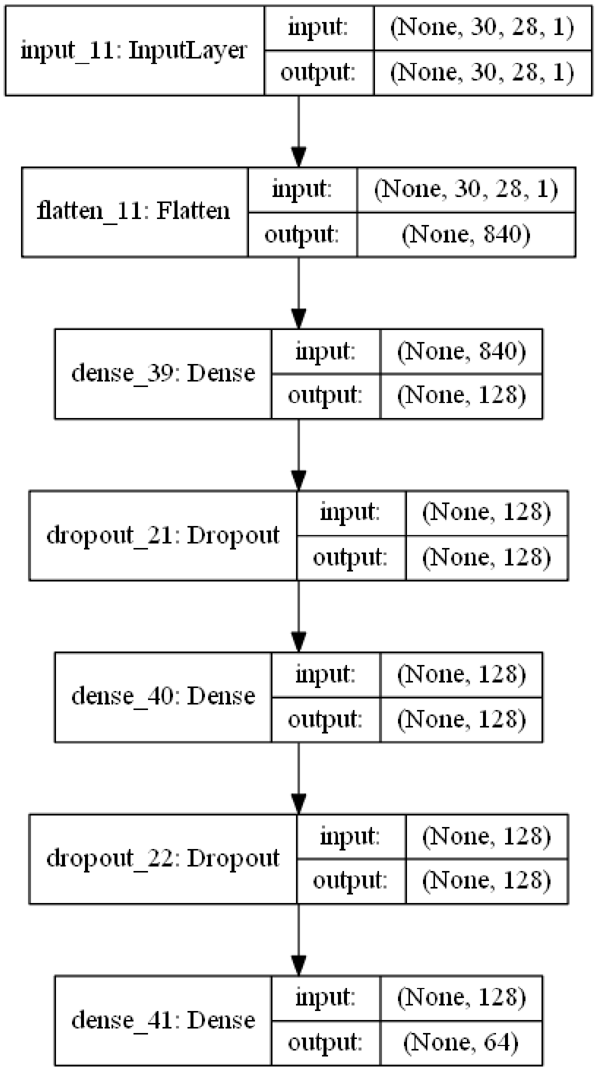

Figure 7: Base neural network architecture

To check the result of this triplet loss function, a visualization of data before and after training is created and distribution of points for every class is observed. Base layer of triplet loss network gives an output of 64-dimensional vector.Since it is difficult to visualize 64-dimensional data, PCA is performed on the data. Two PCA componets which covers the most variance of data are considered for visualization. These two components of PCA are visualized in a scatter plot. From Figure8 we can see that distribution of classes after training is clustered.Figure on the left shows us the scattering of two PCA components before training. Plot on the right shows the visualization of PCA components after training. A clear clustering of classes can be seen in the plot of components after training. Legend of the plot represent each subject. in this particular plot subjects 001 to 020 are visualized. These clustered classes can be further used to separate the individual using a machine learning algorithm such as KNN.

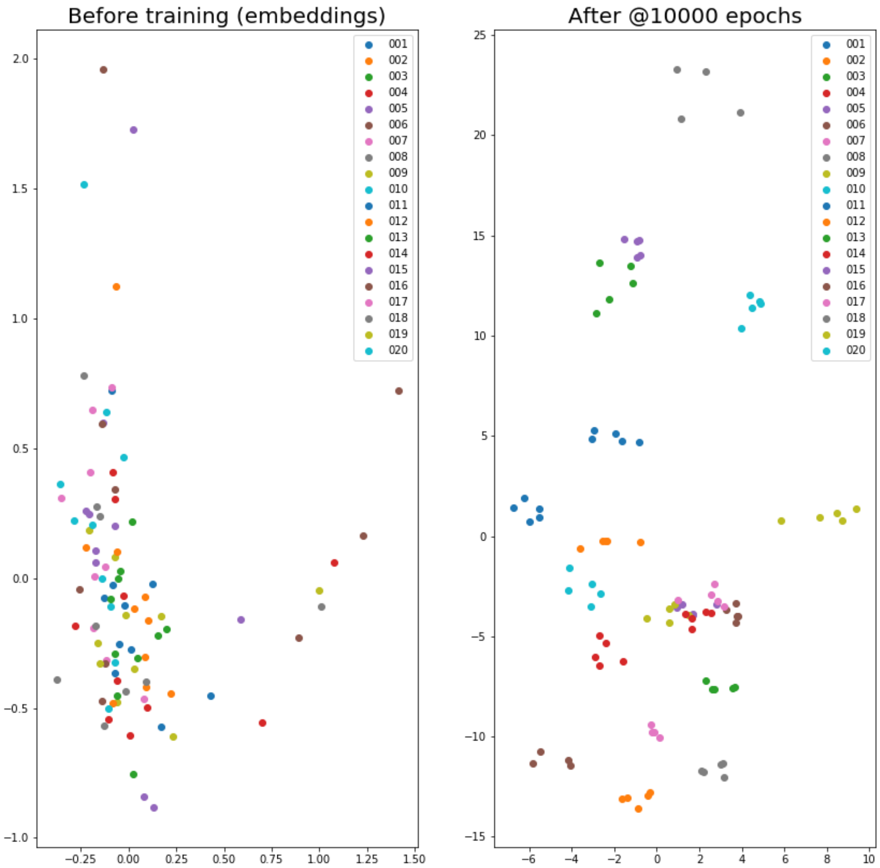

Figure 8: PCA components of embedding output before and after training. Legend shows individual subject number.

5.4 Nearest Neighbour algorithm

The k-nearest neighbour algorithm is applied for identification of individual in both approaches. Manually extracted features cleaned and processed for missing values. Processed features are split into train and test. KNN identifies the individual based on the neighbouring points in the n-dimensional field. Number of neighbour to be considered for identifing persion with less error is decided based on the k value. In order to find the optimum k value, error rates for various k values are calculated. These error rates are plotted as a line chart. Value with least error rate is selected to achieve maximum accuracy. The same process is followed for embedding data in second model. We can see from Figure9 that k=5 for manual features and k=4 for triplet loss features has less error rate respectively. so two models with k=5 for Manual feature and k=4 for triplet loss are created.

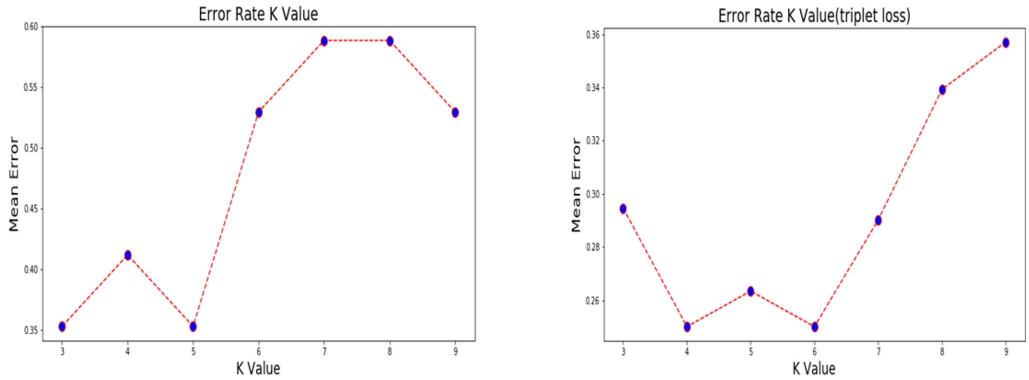

Figure 9: Error rate for various K values for Manual and triplet loss features

6 Evaluation

6.1 Triplet loss architecture

Key point coordinates data is split into train and validation set. Using this data, triplet loss network is trained for 10000 epochs, and the triplet loss for train and validation is noted after every epoch. The value of loss started as 0.8 and reduced to 0.05 after 10000 epoch. Figure10 shows the training loss and validation loss after every epoch. It can be noticed that loss decreases gradually after every epoch.Few peaks in the plot occurred due to jumping of local minima during back propagation algorithm.

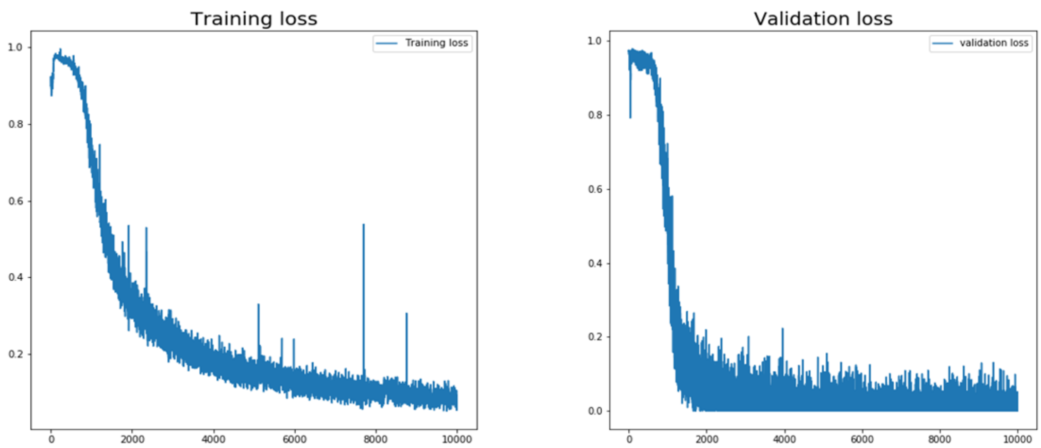

Figure 10: Training and Validation loss for 10000 epoch of triplet loss architecture

6.2 KNN Manual Features

After splitting the manual feature data into train and test, Knn is fitted using train with best k as mentioned in implementation. A classification report for the test data is created. Table 1 shows the accuracy and various other metrics of the classification such as F_1 score, precision. For the

Table 1: Classification report for manual feature method.

Classes	precision	recall	f1-score	support
1	0	0	0	2
2	0.50	1	0.67	2
3	1	1	1	1
4	1	1	1	2
5	1	0.50	0.67	2
6	0.50	0.50	0.5	2
7	1	0.50	0.67	2
8	1	1	1	2
9	0.33	0.5	0.40	2
micro avg	0.65	0.65	0.65	17
macro avg	0.70	0.67	0.66	17
weighted avg	0.69	0.65	0.64	17

purpose of simplicity only 10 classes are considered for report. An average F_1 score of 65% is achieved using manual feature approach.

6.3 KNN triplet loss embedding

After successfully training the triplet loss model, it is used to create the embedding vector of 64 dimension for every video. These embedding along with labels is split into train and test for fitting a Knn algorithm. Knn is fitted with the train data for an optimum K value. The fitted Knn model is evaluated using test data. We can see that from Table2, that accuracy of the triplet loss based knn model is 71%. others metrics of classification such as F_1 score, precision and recall are in the same range.

Table 2: Classification report for Triplet loss- method.

Classes	precision	recall	f1-score	support
116	1	1	1	2
117	1	1	1	2
118	0.50	1	0.67	1
119	0.50	0.50	0.50	2
120	0.50	0.50	0.50	2
121	0.50	0.50	0.50	2
122	1	0.50	0.67	2
123	0.67	1	0.80	2
124	1	0.50	0.67	2
micro avg	0.71	0.71	0.71	17
macro avg	0.74	0.72	0.70	17
weighted avg	0.75	0.71	0.70	17

7 Discussion

In this research Gait recognition is achieved by using OpenPose key points extracted for every frame in a CASIA-B dataset. Two approaches are used for creating a gait recognition namely Manual feature extraction and triplet loss architecture. Manual feature extraction various features for every individuals gait cycle such as length of limbs and angle between limbs and length of stride are calculated. Combination of these features are different for every individual and these features can be used for identification of individual. This similar to the gait energy image used in previous research which is used as a feature for gait cycle silhouettes. An overall accuracy of 65 percent is achieved using this method. A precision rate of macro average 67 percent achieved using this technique. Though the accuracy of this method is comparatively less. This method does not require more data or a machine learning algorithm to extract data. Features extracted from these data are understandable indication of individuals gait. These features can also be used for other gait analysis in medical field. Triplet loss architecture gives us some dynamic features of a persons gait, which are separable from other persons gait. Output features achieved from this method are a 64-dimensional vector. Though these features cluster the classes, these are not understandable indications of persons gait. This process gave a precision score of 71. Using this method reduces the hectic process of creating a Gait energy image which includes aligning and scaling all silhouettes of gait cycle. Further research on this method of gait recognition using OpenPose may lead to near real time gait recognition of individuals.

8 Conclusion and Future Work

There are many researches for improving accuracy Behavioural biometrics because of their advantages over psychological biometrics. Most of the research in Gait recognition are based on silhouettes of every frame in a gait cycle and an aggregated template such as Gait energy image. In this project key points achieved from state of art key point estimation algorithm called OpenPose, are used for building the model. Most famous benmark dataset CASIA-B is used for this project. This data set has videos of 124 subjects in 11 different angles and 3 different modes. Only the normal mode video in 90 degrees angle is considered for this project. Two approaches are followed for building the gait recognition model. In first approach manual features are extracted from the gait cycle such as stride length, max angle between limbs, length of the leg. These features are used to train a nearest neighbour algorithm. In second approach dynamic features are extracted with the help of deep learning triplet loss algorithm. A 64-dimensional vector is created for every video which are used as features for nearest neighbour algorithm. F_1 score of 65 is achieved for the manual feature approach and F_1 score of 76 is achieved using triplet loss algorithm.

In future development, algorithms can be trained on videos in all 11 views and 3 modes. A model for each view can be developed or an invariant view independent model can be developed. Different approach used on silhouettes in previous papers can be applied on OpenPose data. An encoder and deep learning-based similarity check algorithm can be created on key point data for achieving more accuracy. One of the disadvantages of using CASIA-B data is resolution of videos. Different dataset with high resolution videos can be created and used for better estimation of key point using OpenPose. This can help to increase the accuracy of gait recognition model.

References

Andriluka, M., Roth, S. and Schiele, B. (2010). Monocular 3d pose estimation and tracking by detection, *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 623–630.

Ariyanto, G. and Nixon, M. S. (2011). Model-based 3d gait biometrics, *2011 International Joint Conference on Biometrics (IJCB)*, pp. 1–7.

Bobick, A. F. and Davis, J. W. (2001). The recognition of human movement using temporal templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(3): 257–267.

Bulat, A. and Tzimiropoulos, G. (2016). Human pose estimation via convolutional part heatmap regression, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **9911 LNCS**: 717–732.

Cheema, M. S., Eweiwi, A. and Bauckhage, C. (2012). Gait recognition by learning distributed key poses, *2012 19th IEEE International Conference on Image Processing*, pp. 1393–1396.

Chen, Y., Shen, C., Wei, X., Liu, L. and Yang, J. (2017). Adversarial posenet: A structure-aware convolutional network for human pose estimation, *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1221–1230.

Germann, M., Popa, T., Ziegler, R., Keiser, R. and Gross, M. (2011). Space-time body pose estimation in uncontrolled environments, *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pp. 244–251.

Goffredo, M., Bouchrika, I., Carter, J. N. and Nixon, M. S. (2010). Self-calibrating view-invariant gait biometrics, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **40**(4): 997–1008.

Guoying Zhao, Guoyi Liu, Hua Li and Pietikainen, M. (2006). 3d gait recognition using multiple cameras, *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pp. 529–534.

Ju Han and Bir Bhanu (2006). Individual recognition using gait energy image, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(2): 316–322.

Khalid Bashir, Tao Xiang and Shaogang Gong (2008). Feature selection on gait energy image for human identification, *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 985–988.

Kusakunniran, W., Wu, Q., Li, H. and Zhang, J. (2009). Multiple views gait recognition using view transformation model based on optimized gait energy image, *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 1058–1064.

Kusakunniran, W., Wu, Q., Zhang, J., Ma, Y. and Li, H. (2013). A new view-invariant feature for cross-view gait recognition, *IEEE Transactions on Information Forensics and Security* **8**(10): 1642–1653.

Li, Z., Yuan, P., Yang, F. and Li, H. (2017). View-normalized gait recognition based on gait frame difference entropy image, *2017 13th International Conference on Computational Intelligence and Security (CIS)*, pp. 456–459.

Makihara, Y., Adachi, D., Xu, C. and Yagi, Y. (2018). Gait recognition by deformable registration, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 674–67410.

Mansur, A., Makihara, Y., Muramatsu, D. and Yagi, Y. (2014). Cross-view gait recognition using view-dependent discriminative analysis, *IEEE International Joint Conference on Biometrics*, pp. 1–8.

Ramanan, D., Forsyth, D. A. and Zisserman, A. (2005). Strike a pose: tracking people by finding stylized poses, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1, pp. 271–278 vol. 1.

Shen, Q., Tian, C. and Du, L. (2019). Pose-based gait cycle detection, *2019 IEEE 2nd International Conference on Electronic Information and Communication Technology (ICEICT)*, pp. 615–618.

Sokolova, A. and Konushin, A. (2019). Pose-based deep gait recognition, *IET Biometrics* **8**(2): 134–143.

Tang, W., Yu, P. and Wu, Y. (2018). Deeply Learned Compositional Models for Human Pose Estimation, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **11207 LNCS**: 197–214.

Wei, S., Ramakrishna, V., Kanade, T. and Sheikh, Y. (2016). Convolutional pose machines, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4732.

Wu, Z., Huang, Y., Wang, L., Wang, X. and Tan, T. (2017). A comprehensive study on cross-view gait based human identification with deep cnns, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(2): 209–226.

Yaacob, N. I. and Tahir, N. M. (2012). Feature selection for gait recognition, *2012 IEEE Symposium on Humanities, Science and Engineering Research*, pp. 379–383.

Yoo, J., Hwang, D., Moon, K. and Nixon, M. S. (2008). Automated human recognition by gait using neural network, *2008 First Workshops on Image Processing Theory, Tools and Applications*, pp. 1–6.

Yu, S., Chen, H., Reyes, E. B. G. and Poh, N. (2017). Gaitgan: Invariant gait feature extraction using generative adversarial networks, *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 532–539.